

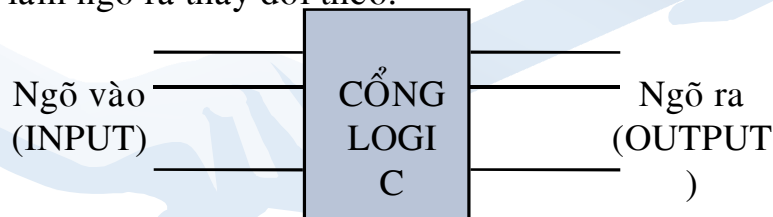
# THIẾT KẾ MẠCH LOGIC TỔ HỢP MSI

## 1. Giới thiệu – Cách thiết kế hệ tổ hợp

Mạch logic được chia làm 2 loại:

- Hệ tổ hợp (Combinational Circuit)
- Hệ tuần tự (Sequential Circuit).

Hệ tổ hợp là mạch mà các ngõ ra chỉ phụ thuộc vào giá trị của các ngõ vào. Mọi sự thay đổi của ngõ vào sẽ làm ngõ ra thay đổi theo.



1

### \* Các bước thiết kế:

- Phát biểu bài toán.
- Xác định số biến ngõ vào và số biến ngõ ra.
- Thành lập bảng giá trị chỉ rõ mối quan hệ giữa ngõ vào và ngõ ra.

<i>Ngõ vào</i>					<i>Ngõ ra</i>			
$X_{n-1}$	...	$X_1$	$X_0$		$Y_{m-1}$	...	$Y_1$	$Y_0$
0	...	0	0					
1	...	1	1					

- Tìm biểu thức rút gọn của từng ngõ ra phụ thuộc vào các biến ngõ vào.
- Thực hiện sơ đồ logic.

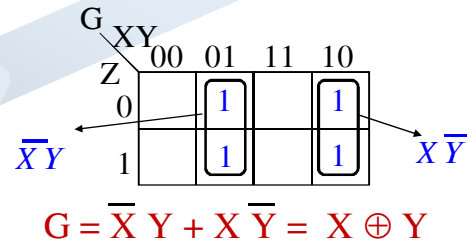
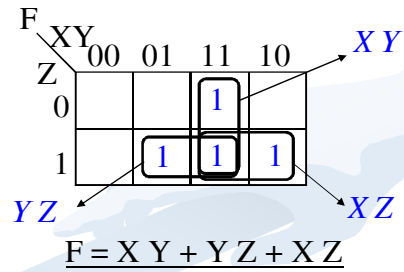
2

**Vd:** Thiết kế hệ tổ hợp có 3 ngõ vào X, Y, Z; và 2 ngõ ra F, G.

- Ngõ ra F là 1 nếu như 3 ngõ vào có số bit 1 nhiều hơn số bit 0; ngược lại F = 0.

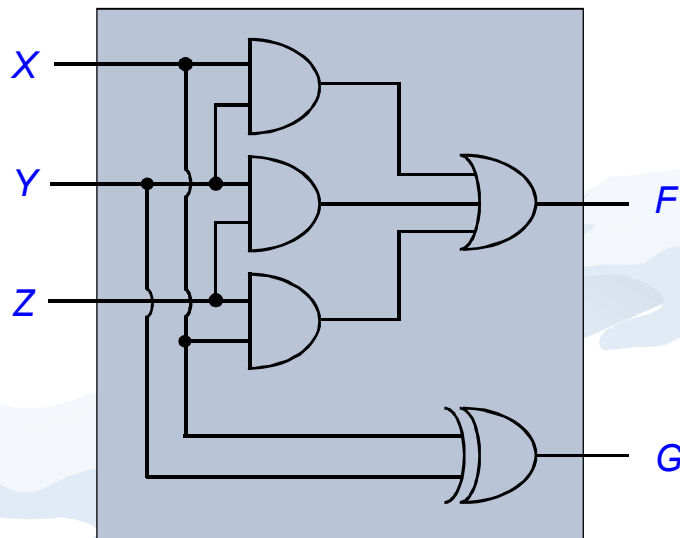
- Ngõ ra G là 1 nếu như giá trị nhị phân của 3 ngõ vào lớn hơn 1 và nhỏ hơn 6; ngược lại G = 0.

X	Y	Z	F	G
0	0	0	0	0
0	0	1	0	0
0	1	0	0	1
0	1	1	1	1
1	0	0	0	1
1	0	1	1	1
1	1	0	1	0
1	1	1	1	0



$F = XY + YZ + XZ$

$G = \bar{X}Y + X\bar{Y} = X \oplus Y$



Trường hợp hệ tổ hợp không sử dụng tất cả  $2^n$  tổ hợp của ngõ vào, thì tại các tổ hợp không sử dụng đó ngõ ra có giá trị tùy định.

**Vd:** Thiết kế hệ tổ hợp có ngõ vào biểu diễn cho 1 số mã BCD. Nếu giá trị ngõ vào nhỏ hơn 3 thì ngõ ra có giá trị bằng bình phương giá trị ngõ vào; ngược lại giá trị ngõ ra bằng giá trị ngõ vào trừ đi 3.

$$F_2 = A + BCD + \bar{B}C\bar{D}$$

$$F_1 = AD + B\bar{C}D + BC\bar{D}$$

$$F_0 = A\bar{D} + B\bar{D} + \bar{A}\bar{B}\bar{C}D$$

A	B	C	D	$F_2$	$F_1$	$F_0$
0	0	0	0	0	0	0
0	0	0	1	0	0	1
0	0	1	0	1	0	0
0	0	1	1	0	0	0
0	1	0	0	0	0	1
0	1	0	1	0	1	0
0	1	1	0	0	1	1
0	1	1	1	1	0	0
1	0	0	0	1	0	1
1	0	0	1	1	1	0
1	0	1	0	X	X	X
1	0	1	1	X	X	X
1	1	0	0	X	X	X
1	1	0	1	X	X	X
1	1	1	0	X	X	X
1	1	1	1	X	X	X

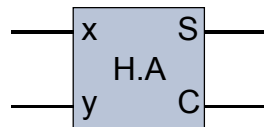
5

## 2. Bộ cộng - trừ nhị phân

### 1. Bộ cộng (Adder):

#### a. Bộ cộng bán phần (Half Adder – H.A):

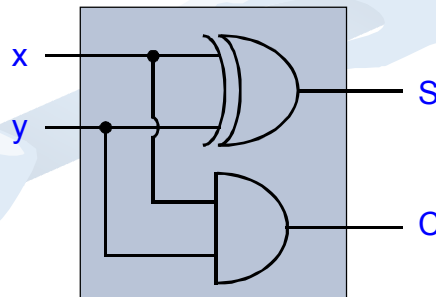
Bộ cộng bán phần là hệ tổ hợp có nhiệm vụ thực hiện phép cộng số học  $x + y$  ( $x, y$  là 2 bit nhị phân ngõ vào); hệ có 2 ngõ ra: bit tổng  $S$  (Sum) và bit nhớ  $C$  (Carry).



x	y	C	S
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

$$S = \bar{x}y + x\bar{y} = x \oplus y$$

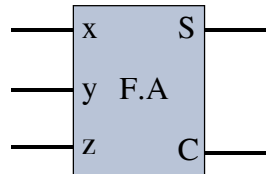
$$C = xy$$



6

### b. Bộ cộng toàn phần (Full Adder – F.A):

Bộ cộng toàn phần thực hiện phép cộng số học 3 bit  $x + y + z$  (z biểu diễn cho bit nhớ từ vị trí có trọng số nhỏ hơn gởi tới)



x	y	z	C	S
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

S	xy	00	01	11	10
z	0		1		1
1		1		1	

$$S = \bar{x}\bar{y}z + \bar{x}y\bar{z} + x\bar{y}\bar{z} + xyz$$

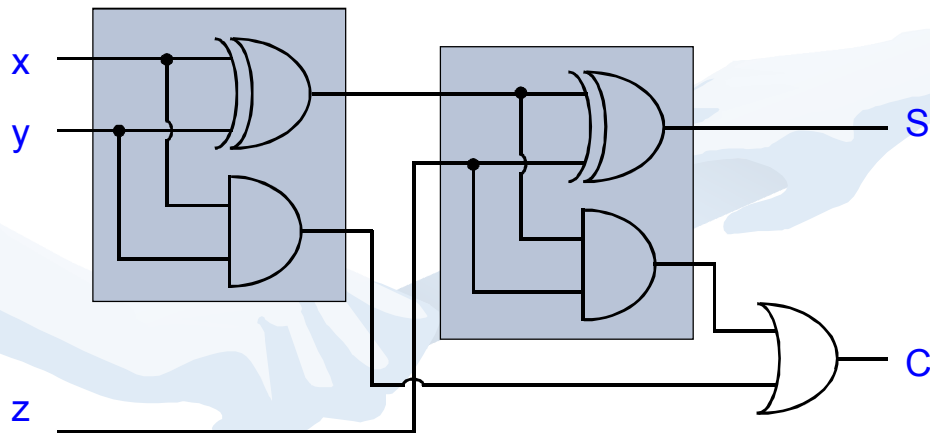
C	xy	00	01	11	10
z	0			1	
1			1	1	1

$$C = xy + xz + yz$$

7

$$\begin{aligned}
 S &= \bar{x}\bar{y}z + \bar{x}y\bar{z} + x\bar{y}\bar{z} + xyz \\
 &= \bar{z}(\bar{x}y + x\bar{y}) + z(\bar{x}\bar{y} + xy) \\
 &= \bar{z}(x \oplus y) + z(\overline{x \oplus y}) \\
 S &= z \oplus (x \oplus y)
 \end{aligned}$$

$$\begin{aligned}
 C &= xy + xz + yz \\
 &= xy + x\bar{y}z + xy\bar{z} + \bar{x}yz \\
 &= xy(1 + z) + z(\bar{x}y + x\bar{y}) \\
 C &= xy + z(x \oplus y)
 \end{aligned}$$

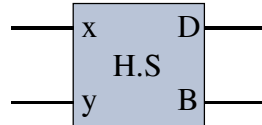


8

## 2. Bộ trừ (Subtractor):

### a. Bộ trừ bán phần (Half Subtractor – H.S):

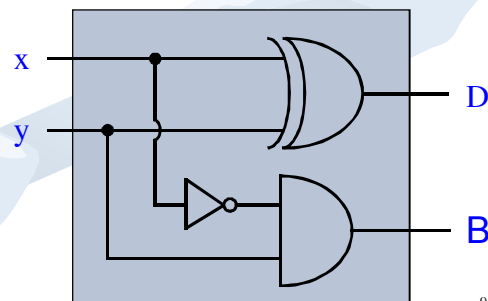
Bộ trừ bán phần có nhiệm vụ thực hiện phép trừ số học  $x - y$  ( $x, y$  là 2 bit nhị phân ngõ vào); hệ có 2 ngõ ra: bit hiệu D (Difference) và bit mượn B (Borrow).



x	y	B	D
0	0	0	0
0	1	1	1
1	0	0	1
1	1	0	0

$$D = \bar{x}y + x\bar{y} = x \oplus y$$

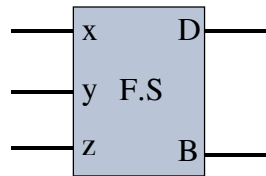
$$B = \bar{x}y$$



9

### b. Bộ trừ toàn phần (Full Subtractor – F.S):

Bộ trừ toàn phần thực hiện phép trừ số học 3 bit  $x - y - z$  ( $z$  biểu diễn cho bit mượn từ ví trị có trọng số nhỏ hơn)



x	y	z	B	D
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	1	0
1	0	0	0	1
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1

D	xy	00	01	11	10
	z	0	1	0	1
	0		1		1
	1	1		1	

$$S = \bar{x}\bar{y}z + \bar{x}y\bar{z} + x\bar{y}\bar{z} + xyz$$

$$S = z \oplus (x \oplus y)$$

B	xy	00	01	11	10
	z	0	1	0	1
	0		1		
	1	1	1	1	

$$C = \bar{x}y + \bar{x}z + yz$$

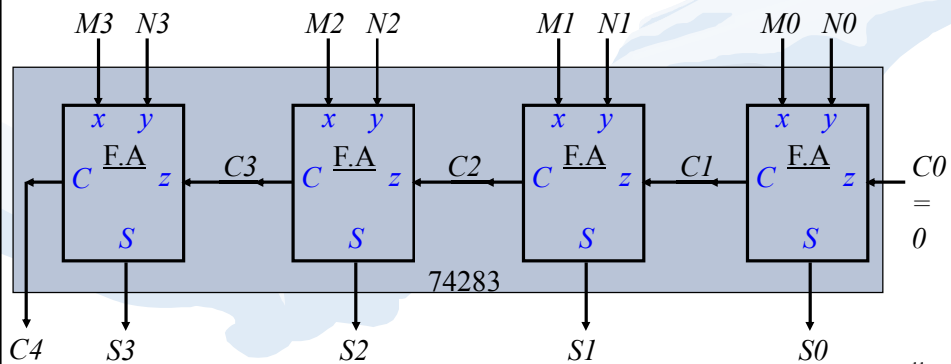
$$C = \bar{x}y + z(\bar{x} \oplus y)$$

10

### 3. Bộ cộng/trừ nhị phân song song:

#### a. Bộ cộng nhị phân:

$$\begin{array}{r}
 \phantom{+} \phantom{C3} \phantom{C2} \phantom{C1} \\
 M: \phantom{C3} M3 \phantom{C2} M2 \phantom{C1} M1 \phantom{C0} M0 \\
 + N: \phantom{C3} N3 \phantom{C2} N2 \phantom{C1} N1 \phantom{C0} N0 \\
 \hline
 C4 \phantom{C3} S3 \phantom{C2} S2 \phantom{C1} S1 \phantom{C0} S0
 \end{array}$$

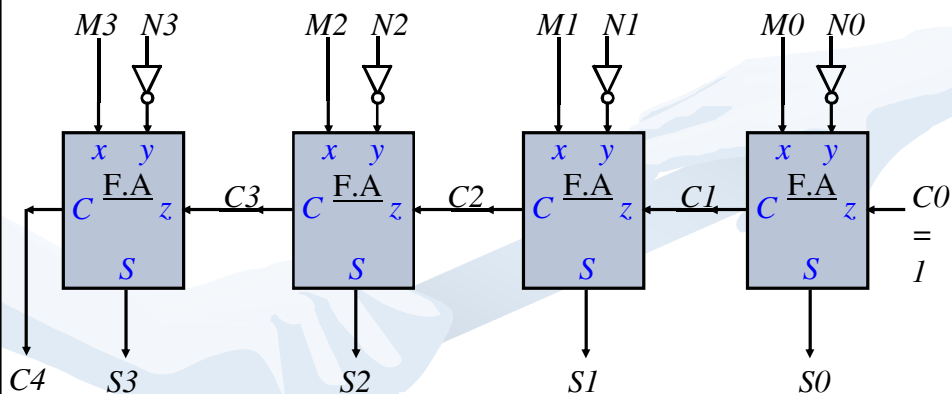


11

#### b. Bộ trừ nhị phân:

- Sử dụng các bộ trừ toàn phần F.S
- Thực hiện bằng phép cộng với bù 2 của số trừ

$$M - N = M + \text{Bù}_2(N) = M + \text{Bù}_1(N) + 1$$



Kết quả: -  $C4 = 1$  kết quả là số dương  
 -  $C4 = 0$  kết quả là số âm

12

### c. Bộ cộng/trừ nhị phân:

Phép toán	$C_0$	$y_i$
CỘNG	0	$N_i$
TRỪ	1	$\overline{N_i}$

Ngõ vào điều khiển

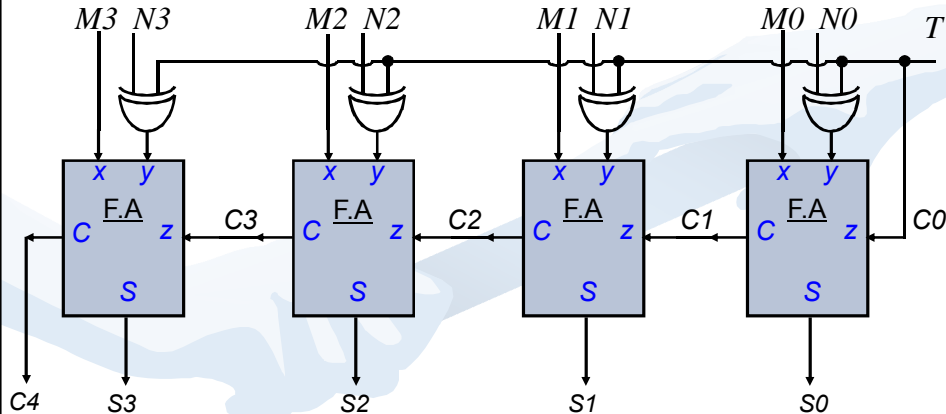
$T = 0$ : Cộng

$T = 1$ : Trừ



$$C_0 = T$$

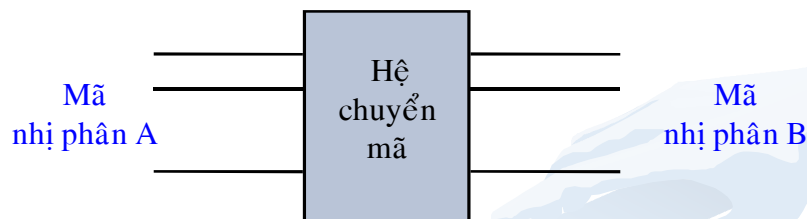
$$y_i = T \oplus N_i$$



13

### 3. Hệ chuyển mã (Code Conversion):

- Hệ chuyển mã là hệ tổ hợp có nhiệm vụ làm cho 2 hệ thống tương thích với nhau, mặc dù mỗi hệ thống dùng mã nhị phân khác nhau.



- Hệ chuyển mã có ngõ vào cung cấp các tổ hợp mã nhị phân A và các ngõ ra tạo ra các tổ hợp mã nhị phân B. Như vậy, ngõ vào và ngõ ra phải có số lượng từ mã bằng nhau.

14

**Vd:** Thiết kế hệ chuyển mã từ mã BCD thành mã BCD quá 3.

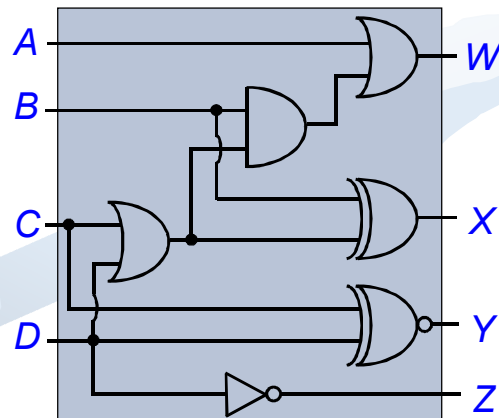
A	B	C	D	W	X	Y	Z
0	0	0	0	0	0	1	1
0	0	0	1	0	1	0	0
0	0	1	0	0	1	0	1
0	0	1	1	0	1	1	0
0	1	0	0	0	1	1	1
0	1	0	1	1	0	0	0
0	1	1	0	1	0	0	1
0	1	1	1	1	0	1	0
1	0	0	0	1	0	1	1
1	0	0	1	1	1	0	0
1	0	1	0	X	X	X	X
1	0	1	1	X	X	X	X
1	1	0	0	X	X	X	X
1	1	0	1	X	X	X	X
1	1	1	0	X	X	X	X
1	1	1	1	X	X	X	X

$$W = A + B(C + D)$$

$$X = B \oplus (C + D)$$

$$Y = \overline{C \oplus D}$$

$$Z = \overline{D}$$

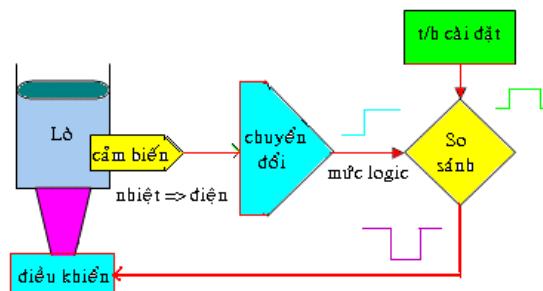


15

## 4. Bộ so sánh độ lớn (Comparator)

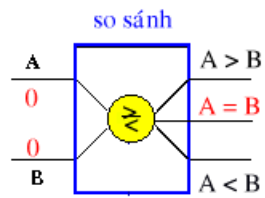
### 4.1. Giới thiệu

- Bộ so sánh là hệ tổ hợp có nhiệm vụ so sánh 2 số nhị phân không dấu A và B (mỗi số n bit).
- Bộ so sánh có 3 ngõ ra ( $A > B$ ), ( $A = B$ ) và ( $A < B$ ); chỉ có 1 ngõ ra tích cực theo kết quả so sánh.





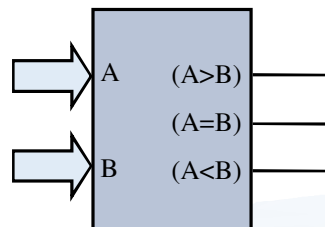
**\* Bộ so sánh 1 bit:**



Ngõ vào		Ngõ ra so sánh		
A	B	A > B	A = B	A < B
0	0	0	1	0
0	1	0	0	1
1	0	1	0	0
1	1	0	1	0

**\* Bộ so sánh 3 bit:**

A:  $A_2 A_1 A_0$   
B:  $B_2 B_1 B_0$



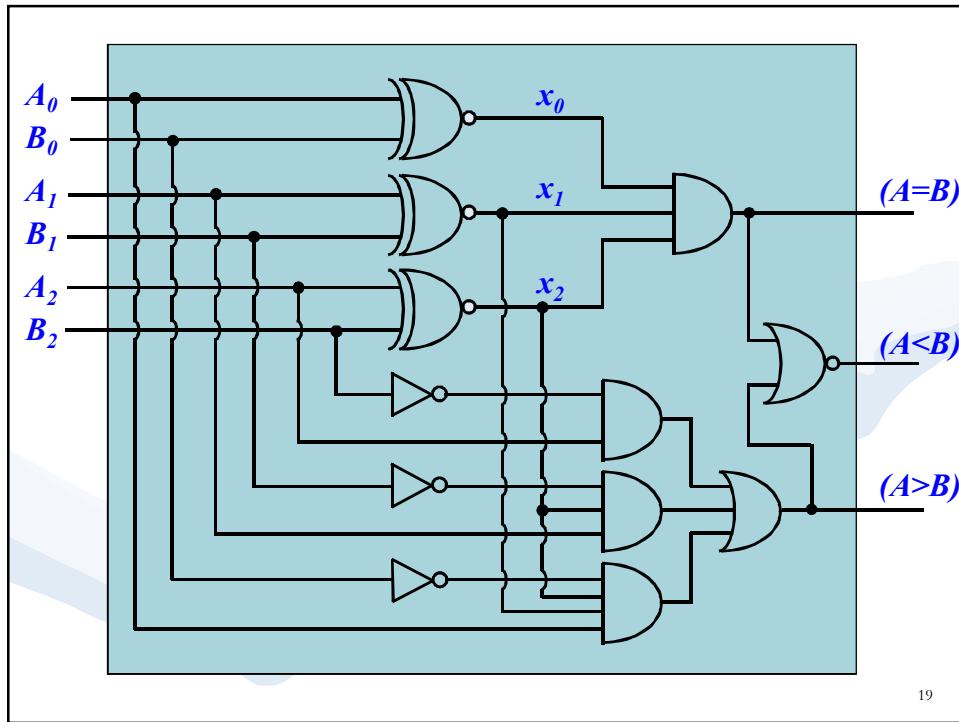
Sử dụng biến trung gian:

$$x_i = \overline{A_i} \oplus \overline{B_i} \quad (i = 0, 1, 2)$$

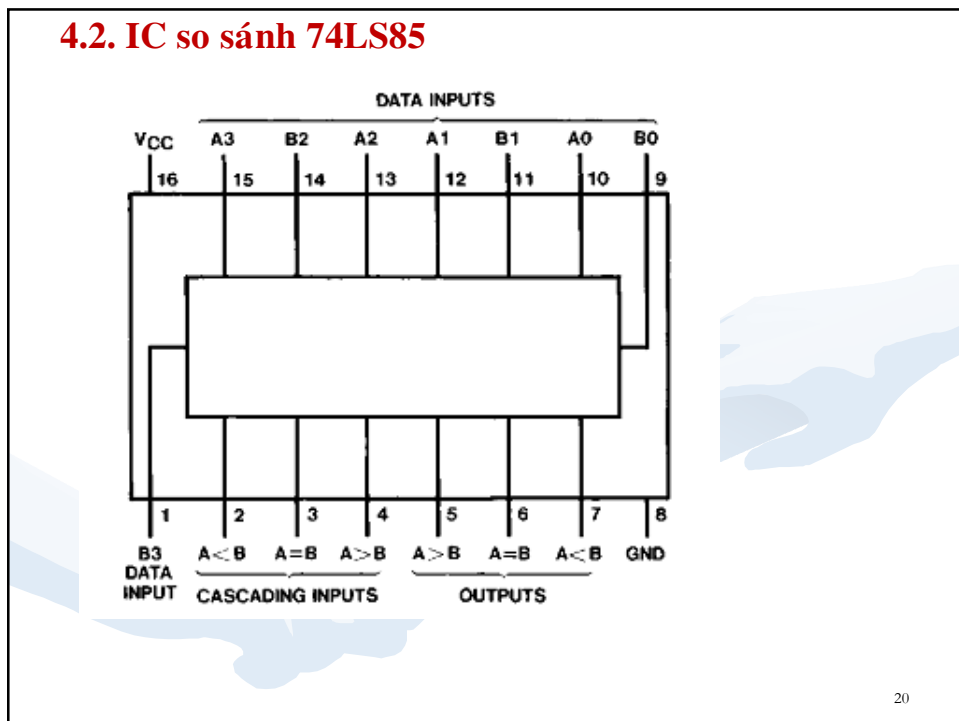
$$(A = B) = x_2 x_1 x_0$$

$$(A > B) = A_2 \overline{B_2} + x_2 A_1 \overline{B_1} + x_2 x_1 A_0 \overline{B_0}$$

$$(A < B) = \overline{A_2} B_2 + x_2 \overline{A_1} B_1 + x_2 x_1 \overline{A_0} B_0 = \overline{(A=B)} + (A > B)$$



## 4.2. IC so sánh 74LS85



## Function Table

Comparing Inputs				Cascading Inputs			Outputs		
A3, B3	A2, B2	A1, B1	A0, B0	A > B	A < B	A = B	A > B	A < B	A = B
A3 > B3	X	X	X	X	X	X	H	L	L
A3 < B3	X	X	X	X	X	X	L	H	L
A3 = B3	A2 > B2	X	X	X	X	X	H	L	L
A3 = B3	A2 < B2	X	X	X	X	X	L	H	L
A3 = B3	A2 = B2	A1 > B1	X	X	X	X	H	L	L
A3 = B3	A2 = B2	A1 < B1	X	X	X	X	L	H	L
A3 = B3	A2 = B2	A1 = B1	A0 > B0	X	X	X	H	L	L
A3 = B3	A2 = B2	A1 = B1	A0 < B0	X	X	X	L	H	L
A3 = B3	A2 = B2	A1 = B1	A0 = B0	H	L	L	H	L	L
A3 = B3	A2 = B2	A1 = B1	A0 = B0	L	H	L	L	H	L
A3 = B3	A2 = B2	A1 = B1	A0 = B0	L	L	H	L	L	H
A3 = B3	A2 = B2	A1 = B1	A0 = B0	X	X	H	L	L	H
A3 = B3	A2 = B2	A1 = B1	A0 = B0	H	H	L	L	L	L
A3 = B3	A2 = B2	A1 = B1	A0 = B0	L	L	L	H	H	L

H = HIGH Level, L = LOW Level, X = Don't Care